

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S52	14195	call with (entry exit)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/23 07:37
S53	549	(717/130,158).CCLS.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/12/23 07:37
S54	129	S52 and S53	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/23 07:37
S56	292	call with (entry exit) with (redundan\$2 same eliminat\$3 remov\$3)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/23 07:48
S57	1	S54 and S56	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/23 07:38
S58	88	call with (entry exit) with (probe instrument\$5)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/23 07:48
S59	28	S53 and S58	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/23 07:41
S60	2	("6,088,525").PN.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/12/23 07:41
S61	13	(call with (before prior)) same ((entry exit) with (method function)) same (probe instrument\$5)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/23 07:49

S62	25501	(eliminate remove) near3 (redundant repeated "same")	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/23 08:09
S63	190	(eliminate remove) near3 (redundant repeated "same") with (probe instrument\$5)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/23 08:09
S64	1	S53 and S63	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/23 08:19
S65	20	call near2 entry near2 "same"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/23 08:21
S66	61	call near2 entry with flow	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/23 08:21
S67	5	S53 and S66	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/23 11:04
S68	36	basic adj block with call with entry	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/23 11:05
S69	549	(717/130,158).CCLS.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/12/23 11:06
S70	3	S68 and S69	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/23 11:36
S71	5	(instrument\$5 probe) near3 prior near3 call	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/23 12:31

S72	70	call near3 entry with flow	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/23 12:32
S73	18	call near3 entry with control adj flow	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/23 12:37
S74	7297	call same (entry enter) same return	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/23 12:37
S75	776	call same (entry enter) same return same exit	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/23 12:37
S77	28	S69 and S75	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/23 13:32
S78	141	(remov\$2 delet\$3 eliminat\$3) near2 redundant near2 (code probe instrument\$5)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/23 13:37
S79	4	S69 and S78	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/23 13:35
S80	2	(remov\$2 delet\$3 eliminat\$3) near2 redundant near2 (code probe instrument\$5) same call	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/23 13:37
S81	10	(remov\$2 delet\$3 eliminat\$3) near2 redundant near2 (probe instrument\$5)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/23 13:39
S82	6	(remov\$2 delet\$3 eliminat\$3) near2 redundant near2 (probe instrumentation)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/23 13:39

S83	0	(remov\$2 delet\$3 eliminat\$3) near2 redundant near2 (instrumentation)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/23 13:44
S84	15	"call block" same "entry block"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/23 13:44


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

 Terms used **probe call redundant instrumentation**

Found 4,220 of 169,166

Sort results by


[Save results to a Binder](#)
[Try an Advanced Search](#)
[Try this search in The ACM Guide](#)

Display results


[Search Tips](#)
☐ Open results in a new window

Results 1 - 20 of 200

 Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

 Relevance scale ☐ ☐ ☐ ☐ ☐

1 [Static and dynamic analysis: Efficient and precise dynamic impact analysis using](#)


[execute-after sequences](#)

Taweessup Apiwattanapong, Alessandro Orso, Mary Jean Harrold

 May 2005 **Proceedings of the 27th international conference on Software engineering**

Publisher: ACM Press

 Full text available: pdf(200.96 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

As software evolves, impact analysis estimates the potential effects of changes, before or after they are made, by identifying which parts of the software may be affected by such changes. Traditional impact-analysis techniques are based on static analysis and, due to their conservative assumptions, tend to identify most of the software as affected by the changes. More recently, researchers have begun to investigate dynamic impact-analysis techniques, which rely on dynamic, rather than static, in ...

Keywords: dynamic analysis, impact analysis, software maintenance

2 [TraceBack: first fault diagnosis by reconstruction of distributed control flow](#)



Andrew Ayers, Richard Schooler, Chris Metcalf, Anant Agarwal, Junghwan Rhee, Emmett Witchel

 June 2005 **ACM SIGPLAN Notices , Proceedings of the 2005 ACM SIGPLAN conference on Programming language design and implementation PLDI '05**, Volume 40 Issue 6

Publisher: ACM Press

 Full text available: pdf(347.77 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Faults that occur in production systems are the most important faults to fix, but most production systems lack the debugging facilities present in development environments. TraceBack provides debugging information for production systems by providing execution history data about program problems (such as crashes, hangs, and exceptions). TraceBack supports features commonly found in production environments such as multiple threads, dynamically loaded modules, multiple source languages (e.g., Java ...

Keywords: fault diagnosis, instrumentation

3 [Scalable cross-module optimization](#)



Andrew Ayers, Stuart de Jong, John Peyton, Richard Schooler

 May 1998 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1998 conference on Programming language design and implementation PLDI '98**, Volume 33